AMENDMENTS TO THE SPECIFICATION

Please delete the section entitled "SUMMARY OF THE INVENTION" in its entirety and substitute the following section therefor:

SUMMARY OF THE INVENTION

In one embodiment, the present invention contemplates a multi-streaming microprocessor core, for executing instruction streams running within the multi-streaming microprocessor core at any time. The multi-streaming microprocessor core includes instruction queues, a bypass structure, and address matching logic. The instruction queues each correspond to each of the instruction streams. The each of the instruction queues has a read pointer, a write pointer, first instructions, store instructions, and load instructions. The read pointer points to an oldest instruction that has not yet been dispatched. The a write pointer points to a newest valid instruction. The first instructions are for dispatch to one or more functional units. The store instructions are for dispatch to a data cache, wherein the store instructions direct write operations. The load instructions are for dispatch to the data cache, where the load instructions direct read operations. Each of the instruction queues retains up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. The bypass structure is ~~within~~ coupled to the data cache. The bypass structure receives the store instructions and has multiple elements. If the write operations hit in the data cache, data corresponding to the write operations are stored in one or more of the elements in the bypass structure before the data is written to the data cache. The address matching logic is coupled to the bypass structure~~ within the data cache~~. The address matching logic receives the load instructions, where the read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache.

An alternative embodiment of the present invention comprehends a multi-streaming microprocessor core, for executing instruction streams running within the multi-streaming microprocessor core at any time. The multi-streaming microprocessor core has instruction queues, a bypass structure, address matching logic, and switching logic. The instruction queues each correspond to each of the instruction streams. The each of the instruction queues ~~include~~ includes a read pointer, a write pointer, first instructions, store instructions, and load instructions. The read pointer points to an oldest instruction that has not yet been dispatched. The a write pointer points to a newest valid instruction. The first instructions are for dispatch to one or more functional units. The store instructions are for dispatch to a data cache, where the store instructions direct write operations. The load instructions are for dispatch to the data cache, where the load instructions direct read operations. Each of the instruction queues retains up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. The bypass structure is ~~within~~ coupled to the data cache. The bypass structure receives the store instructions. The bypass structure has multiple elements, where, if the write operations hit in the data cache, data corresponding to the write operations are stored in one or more of the elements in the bypass structure before the data is written to the data cache. The address matching logic is coupled to the bypass structure ~~within the data cache~~. The address matching logic receives the load instructions, where the read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache. The switching logic is coupled to the bypass structure within the data cache. The switching logic determines where a newest version of the more recent data resides based on bytes, and where one of the read operations matches on multiple elements of the bypass structure.

The present invention further is embodied as a method for eliminating stalls in read and write operations to a data cache within a multi-streaming microprocessor core. The method includes providing multiple instruction streams to corresponding instruction queues. The providing includes within each of the corresponding instruction queues, first pointing to an oldest instruction that has not yet been dispatched; within each of the corresponding instruction queues, second pointing to a newest valid instruction; within each of the corresponding instruction queues, first dispatching first instructions to one or more functional units; within each of the corresponding instruction queues, second dispatching store instructions to a data cache, wherein the store instructions direct write operations; within each of the corresponding instruction queues, third dispatching load instructions to the data cache, wherein the load instructions direct read operations; and within each of the corresponding instruction queues, retaining up to 8 instructions already dispatched so that they can be dispatched again in the case that a short backward branch is encountered. The method also includes first receiving the store instructions in a bypass structure that is coupled to the data cache, where the bypass structure comprises multiple elements, and where, if the write operations hit in the data cache, storing data corresponding to the write operations in one or more of the elements in the bypass structure before the data is written to the data cache.~~first dispatching first instructions to one or more functional units; second dispatching store instructions to a data cache, where the store instructions direct write operations; and third dispatching load instructions to the data cache, where the load instructions direct read operations. The method also includes first receiving the store instructions in a bypass structure within the data cache, where the bypass structure comprises multiple elements, and where, if the write operations hit in the data cache, storing data corresponding to the write operations in one or more of the elements in the bypass structure before the data is written to the data cache.~~ The method further includes second receiving the load instructions in address matching logic~~ within the data cache~~, where the read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache.

Please delete the section entitled "ABSTRACT OF THE DISCLOSURE" in its entirety are substitute the following section therefor:

ABSTRACT OF THE DISCLOSURE

~~Apparatus and method are provided for eliminating stalls in read and write operations to a data cache within a multi-streaming microprocessor core. The apparatus provides a multi-streaming microprocessor core, for executing instruction streams running within the multi-streaming microprocessor core at any time.~~A ~~The~~ multi-streaming microprocessor core is provided that includes instruction queues, a bypass structure, and address matching logic. Each of ~~T~~the instruction queues~~ each correspond to each of the~~

~~instruction streams~~ retains up to 8 instructions already dispatched so that they can be dispatched again. ~~The each of the instruction queues has first instructions, store instructions, and load instructions. The first instructions are for dispatch to one or more functional units. The store instructions are for dispatch to a data cache, wherein the store instructions direct write operations. The load instructions are for dispatch to the data cache, where the load instructions direct read operations.~~ The bypass structure is ~~within the~~ coupled to a data cache. The bypass structure receives ~~the store~~ store instructions and has multiple elements. If ~~the write~~write operations hit in the data cache, data corresponding to the write operations are stored in one or more of the elements in the bypass structure before the data is written to the data cache. The address matching logic is coupled to the bypass structure~~ within the data cache~~. The address matching logic receives ~~the load~~ load instructions, where ~~the read~~ read operations use the address matching logic to search the elements of the bypass structure to identify and use any one or more of the elements representing more recent data than that stored in the data cache.